I claim:

1. A method for reducing a program, M, that preserves at least one branching time property, f, comprising the steps of:

forming a product of said program, M and said branching time property, f, expressed as an automaton, f;

obtaining an abstract domain containing a set of abstract values to generalize possible states of said program and abstract relations that relate said program states to said abstract domain;

computing an abstract program with a reduced number of states and an altered version of said branching time property, f, using said product.

2. The method of claim 1, further comprising the step of performing an automated program check.

3. The method of claim 2, wherein said automated program check is a model checking step.

4. The method of claim 3, wherein said automated program check is performed for an altered branching time property.

5. The method of claim 1, wherein said computing step further comprises the step of defining a set of states, S', in said abstract program as $S' = \overline{S} \times Q$, where S is a set of states in said program, M, and Q is a finite set of states.

6. The method of claim 5, wherein OR states in said set of states, S', are those states where $\delta(q, true)$ has the form $q_1 \vee q_2$ or $\langle a \rangle q_1$, and all other states are AND states, where q are individual states and δ is a transition relation between states.

7. The method of claim 5, wherein an abstract state $(t,\hat{q})$ is in a subset of initial states, I', of the abstract program if there exists $s \in I$ for which $s\xi_{\hat{q}}t$, where s is an individual state, I is a subset of initial states, I, of the program, M, and $\xi_{\hat{q}}$ is one of said abstract relations.

8. The method of claim 5, wherein for an abstract AND state (t, q), the transition $((t, q); (t',q'))$ is in an abstract transition relation, R', if there exists a concrete state (s, q) and a successor (s',q') that are related to (t, q); (t',q') respectively.

9. The method of claim 5, wherein for an abstract OR state (t, q), the transition $((t, q); (t',q'))$ is in an abstract transition relation, R', only if for every (s, q) which is related to (t, q), there exists a successor (s',q') which is related to (t',q').

10. The method of claim 8, wherein said product ATS $M \times A$ is abstracted by weakening said transition relations at AND states.

11. The method of claim 9, wherein said product ATS $M \times A$ is abstracted by strengthening said transition relations at OR states.

12. The method of claim 8, further comprising the step of obtaining one or more rank functions and employing said one or more rank functions in an abstract transition relation, R'.

13. The method of claim 8, further comprising the step of obtaining one or more choice predicates and employing said one or more rank functions in an abstract transition relation, R'.

14. A system for reducing a program, M, that preserves at least one branching time property, f, comprising:

a memory; and

a processor operatively coupled to said memory, said processor configured to:

form a product of said program and said branching time property;

obtain an abstract domain containing a set of abstract values to generalize possible states of said program and abstract relations that relate said program states to said abstract domain;

compute an abstract program with a reduced number of states and an altered version of said branching time property using said product.

15. The system of claim 14, wherein said processor is further configured to perform an automated program check.

16. The system of claim 15, wherein said automated program check is a model checking step..

17. The system of claim 16, wherein said automated program check is performed for an altered branching time property.

18. The system of claim 14, wherein said processor is further configured to define a set of states, S', in said abstract program as $S' = \overline{S} \times Q$, where S is a set of states in said program, M, and Q is a finite set of states.

19. The system of claim 18, wherein OR states in said set of states, S', are those states where $\delta(q, true)$ has the form $q_1 \vee q_2$ or $\langle a \rangle q_1$, and all other states are AND states, where q are individual states and δ is a transition relation between states.

20. The system of claim 18, wherein an abstract state $(t, \hat{q})$ is in a subset of initial states, I', of the abstract program if there exists $s \in I$ for which $s \xi_{\hat{q}} t$, where s is an individual state, I is a subset of initial states, I, of the program, M, and $\xi_{\hat{q}}$ is one of said abstract relations.

21. The system of claim 18, wherein for an abstract AND state (t, q), the transition $((t, q); (t', q'))$ is in an abstract transition relation, R', if there exists a concrete state (s, q) and a successor (s', q') that are related to $(t, q); (t', q')$ respectively.

22. The system of claim 18, wherein for an abstract OR state (t, q), the transition $((t, q); (t', q'))$ is in an abstract transition relation, R', only if for every (s, q) which is related to (t, q), there exists a successor (s', q') which is related to (t', q').

23. The system of claim 21, wherein said product ATS $M \times A$ is abstracted by weakening said transition relations at AND states.

24. The system of claim 22, wherein said product ATS $M \times A$ is abstracted by strengthening said transition relations at OR states.

25. The system of claim 21, further comprising the step of obtaining one or more rank functions and employing said one or more rank functions in an abstract transition relation, R'.

26. The system of claim 21, further comprising the step of obtaining one or more choice predicates and employing said one or more rank functions in an abstract transition relation, R'.